Automatic stimulation of experiments and learning based on prediction failure recognition

Citation for published version (APA):

Juarez Cordova, A. G., Kahl, B., Henne, T., & Prassler, E. (2009). Automatic stimulation of experiments and learning based on prediction failure recognition. In *Proceedings of 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan* (pp. 3607-3612). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/ROBOT.2009.5152285>

DOI:

[10.1109/ROBOT.2009.5152285](https://doi.org/10.1109/ROBOT.2009.5152285)

Document status and date:

Published: 01/01/2009

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Automatic Stimulation of Experiments and Learning Based on Prediction Failure Recognition

Alex Juarez¹, Björn Kahl, Timo Henne and Erwin Prassler²

Abstract—In this paper we focus on the task of automatically and autonomously initiating experimentation and learning based on the recognition of prediction failure. We present a mechanism that utilizes conceptual knowledge to predict the outcome of robot actions, observes their execution and indicates when discrepancies occur. We show how this mechanism was applied to a robot that learns using the paradigm of learning by experimentation, and present first results obtained from this implementation.

I. INTRODUCTION

Consider a robot that is placed in a nearly empty room with an exit in one of the walls blocked by some boxes. If the robot is commanded to go out of the room, it will be able to do so in no time, given that it has enough knowledge of the environment. However, if the robot has only limited or no knowledge about how to get out, how to move boxes, or what it means that there are objects in front of the exit, it will soon face situations where its immediate goals cannot be achieved, and where it will not be able to accomplish its commanded task. In a scenario like this, a good solution for the robot seems to be to improve its performance by learning from its interaction with the environment. For example, a possible course of action could be that the robot tries to learn which boxes can be moved in order to push them out of the way and go through the exit.

Although this approach is appealing, a critical question arises: *when and how should the robot learn?* A possible strategy could be to continuously and immediately learn all the time from the robot's experiences while it performs its commanded tasks. However, the data collected while performing task executions might not be sufficient for successful learning, since learning would be a side effect in this setting. This raises the question of how to distribute the robot's resources between completing its tasks and learning, as in our setting learning is not an end in itself. An approach might be to have an expert tell the robot when it has to learn. This would, however, severely compromise the desired autonomy of the agent.

As a solution we propose to use the recognition of prediction failures as a trigger for learning by experimentation, thus *intentionally but autonomously*

separating phases of task execution and learning. The basic idea here is that the robot will use its available knowledge to plan its actions and predict their outcome, and compare the expected result with the observed result of its actions. Our premise is that a prediction which fails should not only function as an indicator of the opportunity to complete or refine existing knowledge, or even to learn new concepts that might help the robot in achieving the original goal. In addition, we also expect the prediction failure to contain valuable hints, which can guide the robot in its attempt to autonomously work out the root cause for its failure through targeted experimentation.

Following the example presented before, the robot in the almost-empty room may predict that it can “get out” by just moving towards the exit regardless of the objects in front of it. While executing this action, such a prediction will fail if the exit is blocked. The recognition of such failure can be used to initiate a learning process for the acquisition of new knowledge that enables the robot to distinguish between objects which can be moved, and objects which cannot. The robot can then use this learned property of *movable objects* to determine how to change its environment so that it can exit the room.

In a more formal way, the problem that we address in this paper is that of enabling a robotic learner to autonomously and automatically initiate learning of new concepts and theories, in order to improve its performance when solving problems where only limited and/or incomplete knowledge is available. We will only briefly refer to the learning techniques itself, but concentrate on the mechanism for triggering and guiding the experimentation phase, which should produce targeted data for the final learning step.

We present an approach to solve this problem, based on the recognition of prediction failures during the robot's interaction with the environment. This solution will make use of available conceptual knowledge to predict the outcome of robot actions, and compare these predictions to the observed effect of the action execution. As a testbed for the proposed solution, we use an autonomous robot that acquires new knowledge employing the paradigm of *learning by experimentation*, as it is used in the XPERO project¹, in a scenario similar to the one introduced above.

¹ A. Juarez is with the Department of Industrial Design, Eindhoven University of Technology, The Netherlands; (a.g.juarez.cordova@tue.nl).

² B. Kahl, T. Henne and E. Prassler are with the Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg, Germany; (bjoern.kahl@fh-bonn-rhein-sieg.de)

¹More information about the project and learning by experimentation can be found at <http://www.xpero.org>

II. RELATED WORK

Prediction failure has been studied across disciplines and research domains under different names such as disconfirmed belief update, unexpected event analysis, model failure/revision, and artificial surprise. All these terms refer to the same underlying concept: the study of significant divergences between what was expected to occur, and what is observed, and its use in refining and/or correcting the knowledge that cannot explain the observations.

In robotics, the idea of using prediction failure as an instrument to instigate behaviors is not new either, with the majority of work focusing in modeling such failure as a process for action selection and robot control.

For example, Peters [1] introduces one of the earlier models that explicitly encode prediction failures in perceptual robots aiming to produce a reactive behavior to unexpected events. The failure itself is a two-component phenomenon: an expectation set on the basis of previous experience, and a discrepancy from what is projected to happen and what actually happened.

Macedo and Cardoso have done considerable work in this area centered on the motivational aspect of prediction failure. They have presented a computational model based on a graph representation for events of episodic kind, which was later evolved and incorporated in a motivation system that controls action selection [2], [3]. Among their recent work is a control architecture based on emotions, where prediction failure takes a central role [4].

Other approaches focus in human-robot interaction and the roles played by disconfirmed expectations and failed predictions (termed “surprise”), such as the ones by Breazeal [5] and Velasquez [6]. The former introduced a system that implements drives and emotions (surprise among them), as components to regulate the interaction. The latter proposes a cognitive approach to robot control system based on emotions such as surprise (prediction failure), anger, disgust, etc., and mechanisms of attention and perception.

Few approaches, however, have studied the application of similar mechanisms as a trigger for learning, and even fewer have addressed their use as an instigator for learning of concepts, properties and theories that explain physical phenomena surrounding the robot. This is a surprising insight if we consider the power of prediction failure in the learning process of biological agents, affecting the learner’s performance, and in some cases, determining the course of learning [7].

Among the exceptions to this statement is the work from Oudeyer et al. [8], [9], which introduces a motivation

system that allows active learning. The objective of the system is to learn the optimal sensori-motor activity of the robot, focusing on action selection. To achieve this, the assumption is made that the maximum learning opportunity occurs when the predictions on the action effects show the maximum error. Therefore, the best action can be selected by using this information to learn how to improve the predictions.

The application of this approach to an scenario like the one introduced in this paper is, however, not straightforward. Prediction failure and its recognition is applied to relationships between sensori input and actuator activation, e.g. the relationship between setting a specific motor speed and the rate of decrease of distance between the robot and a toy. While this is certainly of interest in the study of prediction failure and its application to active learning and learning by experimentation, it appears to be insufficient if the robot is supposed to initiate learning that helps to realize that a box can be moved out of the way, in order to get out of the room. Furthermore, the use of available concepts, theories and properties of objects is, in our view, needed to solve this task.

The scarcity of approaches that use more abstract and complex conceptual knowledge to recognize prediction failures, indicates that there is an important gap that needs to be filled if we want the robot to autonomously initiate the experimentation and learning in the scenario previously described.

III. APPROACH

We propose a mechanism that utilizes conceptual knowledge to make predictions about the outcome of the robot’s actions, and uses prediction failures to trigger experimentation that will lead to a refinement of the existing knowledge or to the learning of new concepts. Since our approach aims at initiating a targeted collection of data for batch learning, we avoid the term “active learning”, since this mostly refers to systems that learn continuously.

Figure 1 shows a process diagram that describes our approach: *models* of physical phenomena (e.g. the motion of a box that is being pushed by a robot, the rolling of a ball on the floor, etc.) are translated into *predictions* of the effects of action execution.² The sensor data collected by the robot (*observations*) are compared to the predictions, looking for significant divergences. If a divergence is recognized, an experimentation phase is initiated by exploring a subset of the feature space that is relevant to the prediction failure. The data gathered during experimentation should allow Machine Learning tools to refine or learn a new theory.

²By the term “predictions” we mean numerical relations between observable entities as explained in the remainder of the section.

Fig. 1. Data flow of the prediction failure recognition mechanism

A. Using prediction failure to trigger learning

A key element to enable the use of conceptual knowledge in making predictions and recognizing failures is the knowledge representation to use. The knowledge available to the robot is expressed using *first-order logic* restricted to *Horn clauses*. This representation has been traditionally used in Machine Learning tools such as *Prolog* in problem domains where knowledge is incomplete, and deductive/inductive reasoning is required [10], [11].

In general, the predictions are made by obtaining the set of conditions which are related to the action that the robot is commanded to execute. These conditions are expressed by *rules* that are considered to be *true* given some evidence that validates its definition. The advantage of this assumption is that it allows us to consider part of the rule (the *tail* of a first-order logic clause) as the “evidence” the agent is expected to observe, while the head relates to the command issued for execution. Therefore it follows that when the robot decides to execute a command that is associated to a rule, the robot should be able to predict its outcome by deriving from the tail of the rule what will be observed during the action execution. Note that, as rules can be also based on other rules, the search for the set of rules that form a prediction becomes a recursive process.

The principle behind the recognition of a prediction failure is to compare the observations made during the action execution with the conditions found in the rules (the expectations), and trigger a failure when an expectation is not satisfied. To explain this idea, consider a robot that has been provided with a very preliminary and incomplete model for a theory of egomotion, given by the rule

```
move(Robot, Start, Dist, End) :-
    approxEqual(Start, End).
```

The prediction that the robot can make out of this rule, is that its position at the beginning of the action execution, and the one at the end of it are approximately equal, i.e. it doesn’t move. To compare prediction and observations, the robot needs to transform both the rules and the sensor data into a suitable intermediate representation.

In the case of the rules, we provided the robot with the definition of a set of operators [$\approx$, $\neq$, $>$, $<$], as well as a set of entity types [position, distance, height, weight, object]. Each rule is then transformed into a stack based representation by using reverse polish notation (RPN). This notation represents the tail of the rule as a set of operands and operators. For example, transforming the rule for egomotion (move) into RPN would result in a stack containing two operands of type position (*Start* and *End*) and one operator ($\approx$), (See Figure 2).

Fig. 2. A rule transformed into reverse polish notation.

In the case of the observations, the robot tries to abstract the sensor data into symbols that match the operands in the stack, by using the information available about their type. In our example, the operands are of type position and the first parameter of the rule allows us to infer that it refers to the position of the robot, as opposed as the position of other objects in the world. Therefore, the robot will use two measurements of the position of the robot and apply the operator $\approx$ to them.

If the robot then executes a command to move, and it actually observes that there is a change in its own position, the rule will be rendered invalid and lead to a *prediction failure*.

In any case, once a failure in a prediction is triggered, the recognition mechanism collects information about it, such as the models (concepts) that failed or the predicates and variables that reported a divergence. It puts this data at the disposition of an experimentation mechanism that tries to identify the right subset of features to explore and gather data that will be passed on to Machine Learning tools.

B. Multiple predictions or no predictions

It should be noted that there may be several, more or less accurate models that represent theories about the effects of an action execution. For example, consider the case of a robot with a gripper that tries to lift an object, and that is equipped with the following available knowledge:

```
lift(Object, Start, Height, End) :-
    notTooHeavy(Object),
    add(Start, Height, End).
```

```
lift(Object, Start, Height, End) :-
    approxEqual(Start, End).
```

The first predicate (rule) indicates that the robot will lift an object from an initial position, towards a final position which results from adding a specific height to the starting point, only if the object is not too heavy for the robot. The second predicate states that the robot is not able to lift any object at all, since the final position of the object is approximately the same as the initial one. The robot is then able to make predictions for both cases when it lifts the object, and when it doesn't. A prediction failure is, however, signaled only if *all* of the predictions made by the set of related models fail, which means that the robot has found a novel situation which requires triggering an experimentation and learning phase.

We would like to point out that, when faced with a theory like the one in the example, most state-of-the-art Machine Learning tools will conclude that the knowledge available is incomplete, and will try to come up with a more precise theory. Therefore, though it is indeed possible that the robot might find out such set of rules while experimenting with the world, the theory represented by those rules will eventually be refined and replaced with a more consistent one by the learning mechanism, after more information about the environment is obtained (which might involve several episodes of learning).

On the other hand, one of the characteristics of a system that performs learning and experimentation is precisely that the available knowledge might be limited and/or incomplete. Therefore, the existence of a concept or theory that can predict the effect of the robot actions in the world is not guaranteed.

In this case, a fallback mechanism should be present such that the robot is able to recognize, to a limited extent, other stimuli that may lead to the acquisition of new knowledge. In our approach we use a modified version of the *Bayesian surprise* method introduced in [12], [13], which allows to make predictions on the expected sensor values based on probability distributions associated to sensor variables. This technique relies on a Bayesian update of histograms on possible sensor values to determine the most likely values of sensor variables in the next time window. As a result, the robot is able to recognize unexpected changes in the behavior of the observed data, even in the absence of related conceptual knowledge. However, this recognition will offer very limited information on the cause of the failure, as it can only determine which variable shows an unexpected behavior, but not which other variables might be related, or which theories might be worth exploring.

It is also important to note that the assumptions made in order to predict the outcome of an action might result in the robot creating and using imperfect models of the world. However, this is just natural in the kind of setup that we propose, if we consider that a real robot will not be able to observe the environment in its totality at all times

with its own (limited) sensors. Therefore it will take several learning episodes to produce a model that accounts for both observable and partially observable effects of actions.

Moreover, we believe this provides a useful setup to stimulate robotic experimentation: an imperfect model might be sufficient for the robot to perform its task as long as it is able to explain all its observations. Therefore, only when the model fails the robot “feels” the need to learn something new that helps it understand what went wrong, and then tries to interact with the environment to gather data that enables such learning.

IV. EXPERIMENTAL SETUP

We implemented our mechanism for prediction failure recognition in a showcase similar to the one introduced in Section I, both in simulation and real-world. In the former we used an in-house developed simulator XPERSim [14] while for the latter, we utilized educational Eddy robots [15]. The environment in both cases was a large flat area with no walls around and several objects scattered on it. The objects were boxes of the same size but different color and weight, such that half of them were movable by the robot while the other half were not (see Figure 3). The robot was equipped with a bumper sensor, motor encoders and an on-board camera. Additionally, an overhead camera provided information on the absolute position of the objects and robot with respect to a predefined reference frame.

Fig. 3. The environmental setup in simulation 3(a) and real-world 3(b)

The different modules that implement the prediction failure mechanism are interconnected using the framework for design of experiments presented in [16]. This framework provides the additional tools needed to demonstrate the feasibility of our approach: a *design of experiments* component implements a variant of the operator refinement approach appeared in [17], and a *learning* component that relies on the system HYPER [11], a tool for inductive logic programming (ILP) capable of predicate invention.

The outline of our experiments is the following: the robot is provided with incomplete knowledge about a property of the objects in the environment: movability. The robot is then commanded to move different objects in the environment. The robot will apply the incomplete theory and make predictions about the outcome of the commanded motion. The expected result of our experiment is that the

predictions produced from the incomplete theory will fail for some of the objects in the environment. This failure will automatically trigger a new episode of learning by autonomous experimentation by the robot, resulting in a more complete theory being learned.

The knowledge provided to the robot consists of an incomplete theory that describes the movability of objects, expressed as:

```
move(Object, Start, Dist, End) :-
    approxEqual(Start, End).
```

This predicate states that an object's position stays the same, even if the robot is trying to move it from a start position, for a distance, towards an end position. In other words, this predicate describes *all* objects in the environment as non-movable.³ It must be stressed that, prior and during the execution of our experiment, the robot doesn't know which objects are movable and which are not.

Fig. 4. A trace of the robot trying to move the objects in its environment. The robot succeeds with two of them, the green and black boxes.

In our experiments, the robot is commanded to choose randomly one of the four objects in the environment and try to move it. Figure 4 shows a trace of our experiment, where the robot tries to push randomly the objects in the environment. In all cases where the robot chooses the green or black boxes, the evidence (the observed effect of the action) contradicts the prediction: the start and end positions of the object are significantly different, i.e. the predicate `approxEqual(Start, End)` evaluates to be `false`. Upon recognition of the first failure, the robot automatically interrupts the random pushing sequence and initiates a process of targeted autonomous experimentation.

The robot's experiments use the information available on the action executed by the robot, the features relevant to the prediction failure, and several heuristics to design the appropriate experiments that led to the collection of a significant amount of data related to the event. Using this

³The system will generally (re)probe existing knowledge if the knowledge contradicts commands or (recent) experience. This is necessary to plan experiments that can correct a wrong theory.

data the learning tool produces a refined version of the theory that describes the *movability* property of an object, and accounts for objects that can be moved by the robot, and objects that cannot. The new theory is expressed by:

```
p(Obj) :-
    at(Obj, T1, Pos1),
    at(Obj, T2, Pos2),
    different(Pos1, Pos2).

move(Obj, Start, Dist, End) :-
    approxEqual(Start, End),
    not p(Obj).

move(Obj, Start, Dist, End) :-
    add(Start, Dist, End),
    p(Obj).
```

where `at(Obj, T1, Pos1)` states that an object was at position `Pos1` at time `T1`. The interested reader can find more detailed information on the design of experiments mechanism used in [16], as well as details on the Machine Learning techniques and tools in [18], [19].

V. DISCUSSION

Our aim in this paper was to demonstrate the feasibility of our approach in a *learning by experimentation* environment. Therefore, our experiments focused on a specific task that would lead the robot to recognize prediction failure and with it, initiate the learning by experimentation of new theories. The generality of the approach, however, is affected by several factors: the knowledge representation chosen, the grounding of knowledge in sensor data, and the ability to design targeted experiments that will lead to data collections sufficient for learning new (refined) concepts.

While we successfully showed how our approach allowed us to learn the concept of movability, it is very difficult to compare our approach with other autonomous learning system. First, there are not many systems that try to use symbolic knowledge in a general domain for learning by experimentation. The few learning by experimentation systems [20] [21] [22] that exist don't include a "daily task" beside of learning, and are therefore always in the experimentation mode, having no need to trigger a distinct experimentation phase. For systems which use continuous learning (i.e. don't have the concept of a distinct experimentation phase), like most reinforcement learning based systems and other approaches, again the problem of triggering such a experimentation phase does not arise.

The accuracy of the conceptual knowledge (models) determines the quality of the predictions made by the robot, and has a direct influence on the recognition mechanism. In our approach, we search the prediction space, looking for at least one prediction which matches the observations. This could have unfavorable consequences, especially in cases where there are many theories associated to the same robot

action, and when such theories are mostly incomplete or incorrect. A promising idea to improve the robustness of the matching between prediction and observation is to combine this representation with probabilistic data that reflects the “confidence” in a prediction. In this case the algorithm might try to match initially those predictions for which it has a higher confidence of success, and ignore those whose confidence level is too low. Clearly, the development of these ideas must be accompanied by an analysis on the complexity of the search of the prediction space.

Finally, there exists a strong interdependency between the failure recognition and learning mechanisms. Although the approach is general enough to perform well with the family of learning techniques that utilize relational logic and inference from examples, it is also clear that a change in the learning method might demand a redefinition of the way how predictions are created and compared to the observations made by the robot. Our first results show, however, that our mechanism offers a suitable solution to the problem of triggering learning using conceptual knowledge in a learning by experimentation scenario.

VI. CONCLUSIONS

In this paper we presented a method that utilizes conceptual knowledge to autonomously initiate experimentation and learning based on prediction failure. Available models of physical phenomena expressed in first-order logic are turned into predictions of the outcome of robot actions. These predictions are compared to the observations made by the robot. If a mismatch is recognized, the relevant features to be explored during experimentation and learning are derived from the failed prediction. Finally experimentation is initiated to obtain data that will be used by Machine Learning tools to learn new or refined concepts.

We implemented the mechanisms in a *learning by experimentation* setting, obtaining encouraging first results that demonstrate the feasibility of the approach. However, the presented method still needs to be improved for truly autonomous and open-ended learning, as shown in the discussion.

VII. ACKNOWLEDGEMENTS

This work has been funded by the European Commission’s Sixth Framework Programme under contract no. 029427 as part of the Specific Targeted Research Project XPERO (“Robotic Learning by Experimentation”).

REFERENCES

- [1] M. Peters, “Towards artificial forms of intelligence, creativity, and surprise,” in *Proceedings of the 20th Meeting of the Cognitive Science Society*, 1998, pp. 836–841.
- [2] L. Macedo and A. Cardoso, “Modeling forms of surprise in an artificial agent,” in *Proceedings of the 23rd. Annual Conference of the Cognitive Science Society*, 2001.
- [3] —, “Towards artificial forms of surprise and curiosity,” in *Proceedings of the European Conference on Cognitive Science*, S. Bagnara, Ed., 1999, pp. 139–144. [Online]. Available: citeseer.ist.psu.edu/509729.html
- [4] L. Macedo, A. Cardoso, and R. Reisenzein, “A surprise-based agent architecture,” in *Cybernetics and Systems*, R. Trappl, Ed., vol. 2. Austrian Society for Cybernetics Studies, 2006.
- [5] C. Breazeal, “A motivational system for regulating human-robot interaction,” in *AAAI/IAAI*, 1998, pp. 54–61. [Online]. Available: citeseer.ist.psu.edu/breazeal98motivational.html
- [6] J. Velasquez, “An emotion-based approach to robotics,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.
- [7] S. Chaffar and C. Frasson, “The emotional conditions of learning,” in *FLAIRS Conference*, 2005.
- [8] P.-Y. Oudeyer, F. Kaplan, and V. Hafner, “Intrinsic motivation systems for autonomous mental development,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.
- [9] F. Kaplan and P.-Y. Oudeyer, “Curiosity-driven development,” in *Proceedings of the International Workshop on Synergistic Intelligence Dynamics*, 2006.
- [10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ, 2003.
- [11] I. Bratko, *Prolog Programming for Artificial Intelligence*. Addison Wesley Publishing Company, 2001.
- [12] L. Itti and P. Baldi, “Bayesian surprise attracts human attention,” in *Advances in Neural Information Processing Systems, Vol. 19 (NIPS*2006)*. Cambridge, MA: MIT Press, 2006, pp. 1–8.
- [13] —, “A principled approach to detecting surprising events in video,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, June 2005, pp. 631–637.
- [14] I. Awaad, B. Leon, and R. Hartanto, “Xpersim: A simulator for robot learning by experimentation,” in *Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots*, November 2008.
- [15] M. Reggiani and L. Grespan. (2007, November) Eddy: an educational robot device. [Online]. Available: <http://sourceforge.net/projects/eddy/>
- [16] T. Henne, A. Juarez, M. Reggiani, and E. Prassler, “Towards autonomous design of experiments for robots,” in *Proceedings of the 8th. International Workshop in Cognitive Robotics (ECAI 2008)*, Y. Lesperance, Ed., July 2008, pp. 4–9.
- [17] J. G. Carbonell and A. Gil, “Learning by experimentation: The operator refinement method,” in *Machine Learning: An Artificial Intelligence Approach, Volume III*. Morgan Kaufmann, 1990, pp. 191–213.
- [18] G. Leban, J. Zabkar, and I. Bratko, “An experiment in robot discovery with ilp,” in *Proceedings of the 18th. International Conference on Inductive Logic Programming*, September 2008.
- [19] G. Leban and I. Bratko, “Discovering notions using hyper,” University of Ljubljana, Artificial Intelligence Laboratory, Tech. Rep., February 2008. [Online]. Available: <http://www.ailab.si/gregorl/research/discoveringNotions.pdf>
- [20] P. C.-H. Cheng, “Modelling experiments in scientific discovery,” in *International Joint Conferences on Artificial Intelligence, IJCAI*, 1991, pp. 739–745.
- [21] W.-M. Shen, “Discovery as autonomous learning from the environment,” *Machine Learning*, vol. 12, no. 1, pp. 143–165, 1993.
- [22] Y. Gil, “Learning by Experimentation: Incremental Refinement of Incomplete Planning Domains,” in *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann Pub, 1994.